Rényi entropies as a measure of the complexity of counting problems

Claudio Chamon[1] **and Eduardo R Mucciolo**[2]

E-mail: chamon@bu.edu and mucciolo@physics.ucf.edu
[1] Department of Physics, Boston University, Boston, MA 02215, USA
[2] Department of Physics, University of Central Florida, Orlando, FL 32816, USA

Abstract. Counting problems such as determining how many bit strings satisfy a given Boolean logic formula are notoriously hard. In many cases, even getting an approximate count is difficult. Here we propose that entanglement, a common concept in quantum information theory, may serve as a telltale of the difficulty of counting exactly or approximately. We quantify entanglement by using Rényi entropies $S^{(q)}$, which we define by bipartitioning the logic variables of a generic satisfiability problem. We conjecture that $S^{(q\to 0)}$ provides information about the difficulty of counting solutions exactly, while $S^{(q>0)}$ indicates the possibility of doing an efficient approximate counting. We test this conjecture by employing a matrix computing scheme to numerically solve #2SAT problems for a large number of uniformly distributed instances. We find that all Rényi entropies scale linearly with the number of variables in the case of the #2SAT problem; this is consistent with the fact that neither exact nor approximate efficient algorithms are known for this problem. However, for the negated (disjunctive) form of the problem, $S^{(q\to 0)}$ scales linearly while $S^{(q>0)}$ tends to zero when the number of variables is large. These results are consistent with the existence of fully polynomial-time randomized approximate algorithms for counting solutions of disjunctive normal forms and suggests that efficient algorithms for the conjunctive normal form may not exist.

PACS numbers: 89.70.Eg, 89.70.Cf, 03.67.Mn

AMS classification scheme numbers: 03D15, 68Q15

Keywords: counting, satisfiability, complexity, Rényi entropy, entanglement

1. Introduction

Satisfiability (SAT) is an NP-complete problem that aims at deciding whether there is an n-bit string input that satisfies a Boolean logic formula [1, 2]. An example of a satisfiability problem is Circuit Satisfiability, or CSAT, where a circuit is built with a number of gates that is polynomial in n. While the cost of testing whether a given n-bit string satisfies the circuit is polynomial, finding whether the circuit is satisfiable is a hard problem, and counting the number of satisfying inputs is even harder. The problem of counting satisfying solutions of SAT is known as #SAT [3, 4].

Table 1. Summary of the results for the #2SAT problem.

CNF			DNF		
Rényi entropy scaling	efficient algorithm known?	conjecture verified?	Rényi entropy scaling	efficient algorithm known?	conjecture verified?
$S^{(0)} \propto n$	exact: NO	√	$S^{(0)} \propto n$	exact NO:	√
$S^{(1)} \propto n$	approximate:	√	$S^{(1)} \to 0$	approximate:	√
$S^{(2)} \propto n$	NO		$S^{(2)} \to 0$	YES	

The difference in complexity between finding and counting solutions [5] is clearly illustrated in the case of 2SAT, the problem of satisfiability of logic formulas built using Boolean expressions involving exactly two literals (or bits). While the logic expression for a problem in 2SAT can be determined to be satisfiable or not in polynomial time (2SAT $\in$ P), it is believed that counting the number of *all* satisfying solutions (when they exist) cannot be done efficiently. Indeed, #2SAT is a problem in the class #P-complete, which is also the same class containing #3SAT, although 3SAT is in NP-complete. In other words, even though it is much easier to solve 2SAT than 3SAT, counting the satisfying solutions is as difficult in one problem as in the other. Algorithms for counting exactly the solutions of the #SAT problem exist (see for instance [6, 7]), but they require a number of operations that scales exponentially with n. No polynomial or even subexponential algorithm is known.

The fact that a non-exponential algorithm for counting solutions of a SAT problem is still unknown raises the following question: Is there an entropic principle that limits the efficiency of large-scale counting machines, much as there is one that limits the efficiency of thermal engines? Here we propose the Rényi entanglement entropies as means to quantify the difficulty of counting problems. We test the idea explicitly for the case of #2SAT stated in conjunctive normal form (CNF) and its negation, which is stated in disjunctive normal form (DNF).

The entanglement entropy, a concept of much use in quantum information theory, differs from the thermodynamic entropy. In the case of SAT problems, the usual entropy S tells us about the number of solutions Z, i.e. $S = \log_2 Z$. For example, if instead of counting the number solutions one were asked to present all solutions, it would take a time $O(2^S)$ to do so. But counting the number of solutions is a bit easier, in the sense that one could "compress" the information needed to do the counting without presenting all solutions. The degree of compression of the information needed to do the counting is what we relate below to the Rényi entanglement entropies $S^{(q)}$.

In this article, we show rather generically how one can define the Rényi entropies $S^{(q)}$ associated to problems of Boolean expression satisfiability. We then focus on the particular case of random #2SAT problems, compute the Rényi entropies $S^{(q\to 0)}$ and $S^{(q=2)}$, and, in certain cases, $S^{(q\to 1)}$ as well.

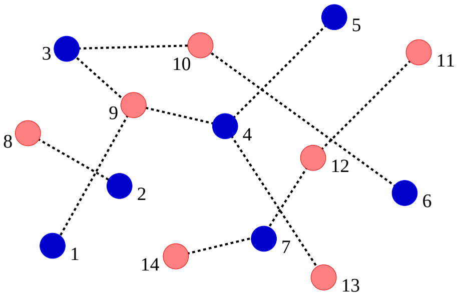

Figure 1. Example of a bipartition of a system of 14 Boolean variables into two sets: A (dark blue) and B (light red). The dashed lines represent 2-literal clauses entering in the Boolean expression of a 2SAT problem.

We conjecture that the Rényi entropy $S^{(q\to 0)}$ provides information on the difficulty of counting solutions exactly, while the Rényi entanglement entropies $S^{(q>0)}$ tell us about the possibility of doing efficient (polynomial-time) approximate counting. More precisely, we conjecture that what determines if the exact counting can or cannot be done efficiently is whether $S^{(q\to 0)}$ scales with n slower or faster than $\log_2 n$. Similarly, what determines whether an approximate counting can or cannot be done efficiently is whether $S^{(q>0)}$ scales with n slower or faster than $\log_2 n$. We then test this conjecture in the case of #2SAT and its negated version using the entropies we obtained numerically for random instances of the problem. We find that in these cases the entropies give the correct prediction for whether efficient algorithms should exist or (likely) not. Our results are summarized in Table 1.

2. Rényi entropies and #SAT

Let us start by considering a set of n binary variables $\{x_j = 0, 1\}_{j=1,\dots,n}$, and a binary weight $W(x) \equiv W(x_1,\dots,x_n) = 0$ or 1, depending on whether the string of binary variables $x \equiv x_1\, x_2 \dots x_n$ satisfies or not a given Boolean expression. With the weights $W(x)$, we define the vector

$$|W\rangle = \sum_{x_n,\dots,x_1=0,1} W(x_1,\dots,x_n)\, |x_1 \dots x_n\rangle, \tag{1}$$

where $|x_1\, x_2 \dots x_n\rangle \equiv |x\rangle$ denotes a particular configuration of this binary system.

Next, we construct the (unnormalized) density matrix

$$\varrho = |W\rangle\langle W| = \sum_{x,x'} W(x)\, W(x')\, |x\rangle\langle x'| \; . \tag{2}$$

The number of strings x satisfying the given Boolean expression is given by

$$Z = \mathrm{tr}\ \varrho = \sum_{x} W(x)^2 = \sum_{x} W(x) \; , \tag{3}$$

where we used that $W(x) = 0, 1$. If one wishes, a normalized density matrix $\rho = \varrho/Z$ can also be constructed.

The entanglement Rényi entropies are constructed from reduced density matrices after dividing the system into subsystems A and B (see figure 1 for an example). Let system A be comprised of bits 1 to l, and system B of bits $l+1$ to n: $x_A \equiv x_1 x_2 \ldots x_l$ and $x_B \equiv x_{l+1} \ldots x_n$, so that $x \equiv x_A x_B$. Next, construct a $2^l \times 2^{n-l}$ matrix $\mathcal{W}_{x_A, x_B} \equiv W(x_A x_B)$ out of the list of weights $W(x)$. One then defines the reduced density matrix

$$\begin{aligned} \varrho_A &\equiv \mathrm{tr}_B\ \varrho \\ &= \sum_{x_A, x'_A} \left(\sum_{x_B} W(x_A x_B)\, W(x'_A x_B) \right) |x_A\rangle\langle x'_A| \\ &= \sum_{x_A, x'_A} [\mathcal{W}\mathcal{W}^\top]_{x_A, x'_A}\ |x_A\rangle\langle x'_A| \,. \end{aligned} \tag{4}$$

The Rényi entanglement entropies [8] are given by

$$S_{AB}^{(q)} = \frac{1}{1-q}\ \log_2 \left[\frac{\mathrm{tr}_A\ \varrho_A^q}{(\mathrm{tr}_A\ \varrho_A)^q} \right] \,. \tag{5}$$

It follows from the cyclicity of the trace that $S_{AB}^{(q)} = S_{BA}^{(q)}$, thus the entropies are independent of the order of the traces (i.e. of which of A or B is traced out first).

The entanglement entropies depend only on the singular values resulting from the decomposition $\mathcal{W} = U\,\Lambda\,V^\top$, where U is an orthogonal $2^l \times 2^l$ matrix, V is an orthogonal $2^{n-l} \times 2^{n-l}$ matrix, and Λ is a $2^l \times 2^{n-l}$ rectangular diagonal matrix with elements $\lambda_k, k = 1, \ldots, d = \min(2^l, 2^{n-l})$. The entanglement entropies are given in terms of these singular values by

$$S_{AB}^{(q)} = \frac{1}{1-q}\ \log_2 \left[\frac{\sum_{k=1}^d \lambda_k^{2q}}{(\sum_{k=1}^d \lambda_k^2)^q} \right] \,. \tag{6}$$

The number of satisfying solutions Z is also linked to the singular values:

$$Z = \sum_{k=1}^d \lambda_k^2 \,. \tag{7}$$

The number of singular values depends on how the system is partitioned. But the partition into two systems is just one step of many: one can further recursively split systems A and B each into two, A_1, A_2, B_1, and B_2, and so on. This is a way to construct a representation of the $W(x)$ as a matrix product state (MPS). The partitions that lead to the largest number of singular values are those when systems A and B are of the same order, so we shall focus on this case for the discussion that follows.

To get the *exact* value of Z, one must sum over *all* the singular values. There are possibly d of them, but many can be zero. So how many are there that are non-zero? The number r of non-zero singular values are related to the Rényi entropy with $q \to 0$, which counts all the non-zero λ_k, all these weighted with the same factor $\lim_{q\to 0} \lambda_k^{2q} = 1$. One finds $\log_2 r = S_{AB}^{(q\to 0)}$.

The rank r sets the size of the matrices that can represent $W(x)$ as a MPS. Therefore, the quantity $S_{AB}^{(q\to 0)}$ is basically a measure of the amount of resources needed

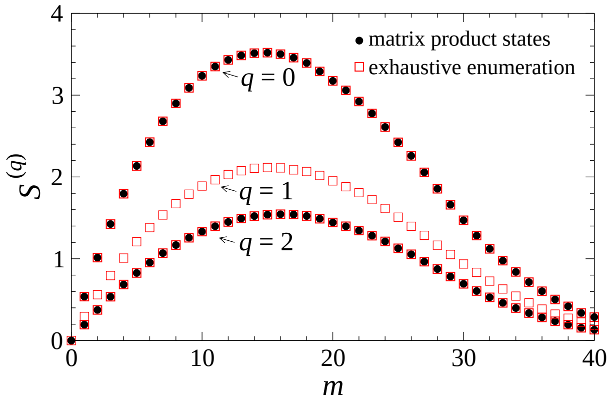

Figure 2. Rényi entropies for the #2SAT problem as functions of the number of clauses for $n = 20$. The data points are averaged over 4000 realizations. The graph provides a comparison between exhaustive enumeration of solutions and the matrix product states method for the cases $q = 0$ and $q = 2$. The case $q = 1$ obtained by exhaustive enumeration is also shown.

to compress the information in all the solutions of the SAT problem. One can show [9] that $r \leq \min(2^l, 2^{n-l}, Z)$, and thus $S_{AB}^{(q\to 0)} \leq S = \log_2 Z$. Because $S_{AB}^{(q\to 0)} \leq S$, the complexity of counting is smaller than that of listing the solutions, for counting could in principle be done by working with the compressed information without the need to expand it.

3. Numerical computations of $S^{(q)}$ for the #2SAT problem

Let us henceforth concentrate on calculating the Rényi entropies for the specific case of #2SAT. We consider random 2SAT problems, where m clauses are drawn uniformly among $n(n-1)/2$ bit pairs, and among the four possible clauses that involve an OR and two literals (each of which can be negated or not). We then compute the entropies $S_{AB}^{(q\to 0)}$ and $S_{AB}^{(q=2)}$. Data for systems of size up to $n = 20$ can be easily obtained from the weights $W(x)$ by exhaustive enumeration of all possible inputs x (see figure 2); however, for larger sizes we deploy a matrix computing scheme based on the method of [10] (details of the method are provided in Appendix A and Appendix B). The results from the matrix computing method for $q = 0$ and $q = 2$ were compared against the results of exhaustive enumeration of solutions for the same random problem, realization by realization, and found to match each other within the relative error margin controlled by the threshold used to distinguish the smallest non-zero singular value from zero (typically, 10^{-10}). In figure 2, we present results using both methods for $n = 20$ after averaging over 4000 realizations.

In figure 3a we show how the entropies per bit $s^{(0)} = S_{AB}^{(q\to 0)}/n$ and $s^{(2)} = S_{AB}^{(q=2)}/n$ vary as we increase the ratio between the number of gates and the number of bits, $\alpha = m/n$. The numerical data was averaged over 4000 realizations of random 2SAT problems. The very weak dependence of the entropies per bit on n lead us to conclude

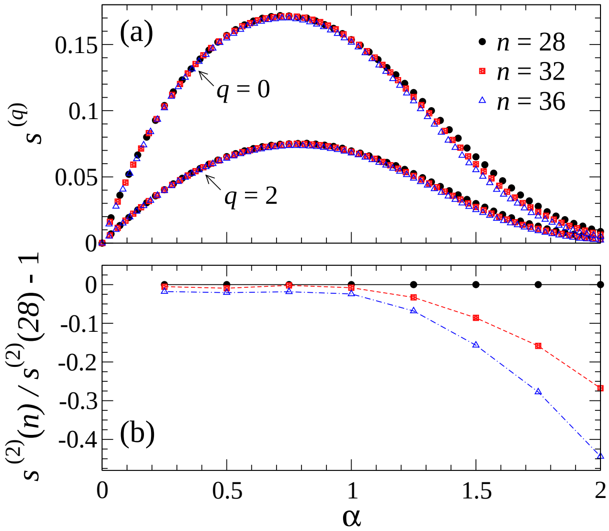

Figure 3. (a) Dependence of two Rényi entropies per bit, $s^{(0)} = S^{(q\to 0)}/n$ and $s^{(2)} = S^{(q=2)}/n$, with the ratio $\alpha = n/m$. The data shows that the entropies per bit are very weakly dependent on n for $\alpha < 1$. The matrix product state method was employed in the simulations and a total of 4000 realizations were used for each data set. (b) The relative deviation $s^{(2)}(n)/s^{(2)}(28) - 1$ as a function of α, showing in more detail the weak dependence of $s^{(2)}$ with n when $\alpha < 1$. The strong deviation seen when $\alpha > 1$ indicates the onset of the 2SAT phase transition at $\alpha = 1$. The lines are guides to the eye.

that both $S_{AB}^{(q\to 0)}$ and $S_{AB}^{(q=2)}$ scale as the "volume", i.e. the entropies are very closely proportional to the system size n up to $\alpha = 1$. The deviation from this linear behavior is very small, less than 3%, as shown in figure 3b, and can be attributed to the finite number of bits in our simulations. A finite-scaling analysis indicates that the entropies per bit saturate to a finite value as $n \to \infty$ when $\alpha < 1$. We note that the value $\alpha = 1$ marks the onset of a well-known phase transition in the random 2SAT problem [11, 12, 13]: for $\alpha > 1$, one expects $S^{(q)}/n \to 0$ as $n \to \infty$ (i.e., the number of solutions goes to zero when $m > n$). This is consistent with the growing deviation from the linear behavior that we observe in figure 3b beyond $\alpha = 1$.

From the perspective of the matrix computing technique we employed in our simulations, the difficulty in counting solutions of a given realization of an m-clause #2SAT problem is not directly related to the value of $S^{(q)}$ computed after the mth clauses. Instead, the difficulty is measured by the maximum value that $S^{(q)}$ reaches as the logic gates are applied, and this maximum can be reached at an intermediate number of clauses m^*, before all m clauses are enforced. Typically, Rényi entropies reach a maximum value around a value $m^* \approx n$ and quickly decay beyond that point (see figure 2). Nevertheless, for all values of m, the averaged Rényi entropies for CNF always scale linearly with n, provided that $\alpha < 1$.

4. The negation version of the #2SAT

Next, we compare and contrast the entanglement entropies computed for the 2SAT problems with those for the negation of the same problems. More precisely, let us consider the Boolean expressions $\overline{W}(x) = 0$ if $W(x) = 1$ and $\overline{W}(x) = 1$ if $W(x) = 0$.

The singular values obtained from the weights $\overline{W}(x)$ are intimately correlated to those derived from the weights $W(x)$. We find empirically that the relation for large n should be as follows (see Appendix C for a heuristic argument) in the case when the number of satisfying solutions is less than the number of non-solutions, i.e. $Z < 2^n/2$. For each and every realization of the problem, if there are r non-zero singular values λ_k^2, $k = 1, \ldots, r$, for the 2SAT problem with weights $W(x)$, then there are $r+1$ singular non-zero singular values $\bar{\lambda}_k^2$, $k = 0, \ldots, r$, for the negated problem with weights $\overline{W}(x)$. The relation between these two sets of singular values, which should hold in the large n limit, is

$$\bar{\lambda}_0^2 = 2^n - 2Z \tag{8a}$$

$$\bar{\lambda}_k^2 = \lambda_k^2, \quad k = 1, \ldots, r\ . \tag{8b}$$

Notice that $\bar{Z} = \sum_{k=0}^{r} \bar{\lambda}_k^2 = 2^n - Z$, which is the number of solutions of the negated problem. To support our claim that the singular values of the two problems are related according to (8*a*,8*b*), we plot in figure 4 the pairs of singular values $(\lambda_k^2, \bar{\lambda}_k^2)$, for $k = 1, \ldots, r$. The data displayed are for $n = 20$ and 1000 realizations of the random 2SAT problem and its negation. We also computed the relative deviation $\epsilon = |\bar{\lambda}_0^2 - (2^n - 2Z)|/(2^n - 2Z)$ for the highest singular value, and averaged this deviation over the 1000 realizations. We find that $\epsilon_{\text{ave}} < 0.3\%$. We conclude that even for n as small as 20 the relations in (8*a*,8*a*) hold rather well.

The relationship between the singular values for W and $\bar{W}$ yields, via (6), a connection between the respective Rényi entropies:

$$2^{(1-q)\bar{S}_{AB}^{(q)}} = \left(Z/\bar{Z}\right)^q \, 2^{(1-q)S_{AB}^{(q)}} + \left(1 - Z/\bar{Z}\right)^q\ . \tag{9}$$

In particular

$$2^{\bar{S}_{AB}^{(q\to 0)}} = 2^{S_{AB}^{(q\to 0)}} + 1\ , \tag{10}$$

as expected since there is one more singular value in the negated problem, and

$$\bar{S}_{AB}^{(q\to 1)} = \left(\frac{Z}{\bar{Z}}\right) S_{AB}^{(q\to 1)} - \left[\left(\frac{Z}{\bar{Z}}\right) \log_2\left(\frac{Z}{\bar{Z}}\right) + \left(1 - \frac{Z}{\bar{Z}}\right) \log_2\left(1 - \frac{Z}{\bar{Z}}\right)\right]\ . \tag{11}$$

For a #2SAT problem with a non-zero ratio α of clauses to variables, the number of satisfying solutions is exponentially smaller than the number of unsatisfying solutions: $\lim_{n\to\infty} Z/\bar{Z} = 0$. Therefore, we find that in this case $\bar{S}_{AB}^{(q>1)} \to 0$ in the large n limit.

5. Rényi entropies and the complexity of #SAT

With these results for the Rényi entropies, we are in a position to test whether or not, at least for the #2SAT and its negation, the entropies can predict the degree of difficult in solving a problem.

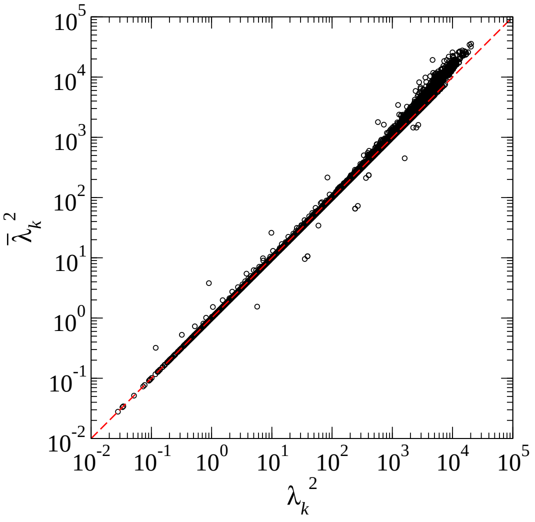

Figure 4. Correlation between the singular values associated to W and to the negation $\bar{W}$ for $n = 20$. For each of 1000 realizations of the problem, the singular values λ_k^2 for the a random 2SAT problem and the $\bar{\lambda}_k^2$ ones associated to the negated problem are rank ordered, and the pair $(\lambda_k^2, \bar{\lambda}_k^2)$ is plotted. The total number of pairs in the plot, for all the 1000 realizations combined, is 14827. The dashed straight line is a guide to the eye. The slight deviations away from the line at higher values of the singular values are dominated by the second largest singular value of the negated problem, the one with $k = 1$.

- *2SAT solutions* – In the case of #2SAT, we find that both $S^{(q\to 0)}$ and $S^{(q=2)}$ are volumetric, i.e. they scale linearly with n. Since $S^{(q\to 0)} \geq S^{(q\to 1)} \geq S^{(q=2)}$, the entanglement entropy with $q \to 1$ is also volumetric. Because $S^{(q\to 0)}$ is volumetric, carrying out the counting exactly cannot be done efficiently, which is expected since, after all, #2SAT is #P-complete and hence unavoidably hard. Thus, in this case, the entropy is a good predictor of the difficult of the problem. Moreover, because $S^{(q\to 1)}$ is also volumetric, we do not expect that an approximate counting algorithm should exist either [4].
- *2SAT non-solutions* – In the case of the negation of #2SAT, we find that $\bar{S}^{(q\to 0)}$ is volumetric, while $\bar{S}^{(q\to 1)}$ vanishes for large n. We would then conclude that counting the exact number of non-satisfying binary strings is still unavoidably hard. However, because $\bar{S}^{(q\to 1)} \to 0$, one can efficiently approximate the number of non-satisfying inputs. This results is in agreement with what is known about the problem, as we explain below in more detail.

That counting exactly in one problem is equivalent to counting exactly in the other is evident, as $\bar{Z} = 2^n - Z$. Thus, if one finds Z one has $\bar{Z}$ and vice versa. But since $\bar{Z} \gg Z$, it is easier to find an approximation to $\bar{Z}$ than to Z within the same *relative* error. The issue can be alternatively stated as follows. A 2SAT problem is presented in conjunctive normal form (CNF), with m OR-clauses with two literals, all AND-ed together. Its negation is then in disjunctive normal form (DNF), with m AND-clauses with two literals, all OR-ed together. It is known that DNF problems admit a fully polynomial-time randomized approximation scheme (FPRAS) [14]. That the entropy $\bar{S}^{(q\to 1)} \to 0$ while, in contrast, $S^{(q\to 1)}$ is volumetric, is the way in which our entanglement entropy approach signals that the DNF problem is simpler than the CNF one.

6. Conclusion

Let us conclude with a few remarks. First, for CNF problems defined on a graph, one may expect that the entanglement should be volumetric. However, it is not always the case that a problem defined on graph will have volumetric entanglement, as exemplified in the DNF problem that results from the negation of a CNF one. The intuition as to why that is so is that the disjunctive form splits the problems into many disjoint ones, whereas the conjunctive ones ties the bits together, and is mean-field like. The entanglement is one way to capture the fact that some problems cannot be simplified by dividing into subproblems, while others can.

Finally, we would like to mention a possible practical way to explore the entanglement as predictor of whether it may or not be possible to approximately count for a given problem. Notice that we succeeded in getting the scaling of the Rényi entropies even for systems of modest sizes. Therefore, one could get a sense of whether a FPRAS may possibly exist, without explicitly constructing one, using an entanglement entropy finite-size scaling analysis.

Acknowledgments

This work was supported in part by the NSF grants CCF-1116590 and CCF-1117241. We would like to thank H. Castillo, C. Laumann, and P. Wocjan for useful discussions.

Appendix A. Numerical simulations with matrix product states

We employed the matrix computing method introduced in [10] to verify numerically that $S^{(q)} \propto n$ for the conjunctive form of the #2SAT problem. In this method, we associate to each binary variable x_j a pair of real matrices M_j^0 and M_j^1 of dimensions $D_{j-1} \times D_j$. The weight $W(x)$ of each configuration x is written as the trace of a product of these matrices, namely,

$$W(x_1, \ldots, x_n) = \mathrm{tr}\left(M_1^{x_1} \cdots M_n^{x_n}\right). \tag{A.1}$$

(The trace can be dropped if we consider the first and last matrices to be row and column vectors, respectively, namely, $D_0 = D_n = 1$.) It is straightforward to show that Z, the number of satisfying configurations, is given by the expression

$$Z = \mathrm{tr}\left[\left(M_1^0 + M_1^1\right) \cdots \left(M_n^0 + M_n^1\right)\right]. \tag{A.2}$$

The matrix representation of the weight $W(x)$ allows us treat (1) similarly to a quantum mechanical superposition state. Then, by operating sequentially on adjacent pairs of matrices $(M_j^{x_j}, M_{j+1}^{x_{j+1}})$, we can simultaneously check the satisfiability of all 2^n instances of a CNF problem.

One starts by setting $W = W_0(x) = 1$ for all values of the n-bit string x. This can be easily implemented by choosing $D_j = 1$ and $M_j^0 = M_j^1 = 1$ for all $j = 1, \ldots, n$, yielding $Z = Z_0 = 2^n$. Each clause C_k in a CNF can eliminate non-satisfying instances of the problem and, consequently, reduce Z. We call these gate operations filters. Thus, starting from an initial weight distribution $W_0(x)$, the vector $|W_0\rangle$ evolves into a state $|W_C\rangle$ by the sequential application of filters that block states that do not satisfy a CNF Boolean expression $C = C_1 \wedge C_2 \cdots \wedge C_m$. Notice that there are four possible types of two-bit OR filter gates, depending on whether the input bits are negated or not.

Since the order in which bits appear in the clauses in C is random and only adjacent bit operations are allowed in matrix computing, matrices have to be moved up and down the bit string. This is done through SWAP gates [10]. These SWAP gates, when combined with other logic gates such as OR, tend to rapidly increase the rank of the matrices. This is a limiting factor of the method, as every gate operation between bits $(j-1, j)$ employs a singular value decomposition which requires $O(D_{j-1}^3)$ floating point operations. On the other hand, the filters, by state elimination, partially compensate the growth in matrix rank and $D_{\max} = \max\{D_j\}$ tend to peak around $m \approx n$. Therefore, for each realization of a CNF, the computational cost of the numerical calculation scales as $O(D_{\max}^3)$.

In order to minimize the number of SWAP operations and limit the growth of matrix ranks, we employ two pre-processing strategies. First, we reorder the bits in the string so that the sum of pair-wise distances between bits in the clauses of the CNF is minimized. This is done through a Monte Carlo sampling with a 100% rejection rate if the new configuration has a higher total distance than the previous one. Typically, 30 sampling steps are used.

Second, we reorder the clauses so that those bits participating in few or no clauses are acted on first. Bits which are no longer required are set to an inactive state by absorbing their bit matrices into matrices of a neighboring bit and replacing their matrices with identity ones. These measures reduce the matrix ranks substantially, which in turn allow us to count exactly all solutions for large CNFs. We find empirically that $\langle D_{\max}^3 \rangle \sim 2^{0.1n}$. This favorable scaling (as compared to the computational cost of the best known algorithm for solving exactly the #2SAT problem, which scales as $2^{0.329n}$ [7]) corresponds to an average behavior and does not apply to hard, worst-case realizations of the CNF.

Appendix B. Bit-string partitioning

Once the complete sequence of m clauses of a CNF is implemented with matrix computing, the partition matrix $\mathcal{W}$ can in principle be obtained in the following way. Consider the bit-string matrix set $\{M_j^{x_j}\}$ resulting from the sequence of Boolean gates.

Following the definition of $\mathcal{W}$, we can write

$$\mathcal{W}_{x_A;x_B} = \sum_{\alpha=1}^{D_{n/2}} A_{x_A;\alpha}\, B_{\alpha;x_B}\,, \tag{B.1}$$

where

$$\begin{aligned} A_{x_1\cdots x_{n/2};\alpha} &= M_1^{x_1}\cdots M_{n/2}^{x_{n/2}} \qquad (B.2)\\ B_{\alpha;x_{n/2+1}\cdots x_n} &= M_{n/2+1}^{x_{n/2+1}}\cdots M_n^{x_n}\,. \qquad (B.3)\end{aligned}$$

This decomposition of $\mathcal{W}$ can be used to compute some Rényi entropies without the need to obtain $\mathcal{W}$ explicitly. While it is straightforward to show that $S^{(q\to 0)} = \log_2 D_{n/2}$, $S^{(q>1)}$ requires an additional manipulation. Combining the singular value decomposition $\mathcal{W} = U\,\Lambda\,V^\top$ and with (B.1), we can write

$$\sum_{k=1}^{D_{n/2}} \lambda_k^{2q} = \mathrm{tr}\left(\Lambda^{2q}\right) = \mathrm{tr}\left[\left(U^\top \mathcal{W}\mathcal{W}^\top U\right)^q\right] = \mathrm{tr}\left[\left(BB^\top A^\top A\right)^q\right], \tag{B.4}$$

where we used the cyclicity of the trace. Finally, combining (6) with (B.4), one finds $S^{(q>1)}$.

The two major advantages in this approach as compared to performing the singular values decomposition of $\mathcal{W}$ is the reduced number of floating point operations (since no singular value decomposition of $\mathcal{W}$ is needed) and the reduced memory allocation for matrix storage. The disadvantage is that one cannot compute $S^{(q\to 1)}$. In practice, this is the only viable numerical approach that we know for finding the scaling of $S^{(q)}$ with n when $n > 20$.

Appendix C. Heuristic argument for (8a,8b)

Because $\overline{W}(x) = 1 - W(x)$, the associated matrix $\overline{\mathcal{W}} = \mathcal{M} - \mathcal{W}$, where $\mathcal{W}$ is the $2^l \times 2^{n-l}$ matrix constructed in the main text and $\mathcal{M}$ is a matrix of the same size with all entries equal to 1. It follows that

$$\begin{aligned}\overline{\mathcal{W}}\,\overline{\mathcal{W}}^\top &= \mathcal{M}\mathcal{M}^\top - \mathcal{W}\mathcal{M} - \mathcal{M}\mathcal{W}^\top + \mathcal{W}\mathcal{W}^\top\\ &= 2^{n-l}\,\mathcal{H} - \mathcal{W}\mathcal{M} - \mathcal{M}\mathcal{W}^\top + \mathcal{W}\mathcal{W}^\top\;, \qquad (C.1)\end{aligned}$$

where we defined the $2^l \times 2^l$ matrix $\mathcal{H}$ with all entries equal to 1. Now,

$$\left[\mathcal{W}\mathcal{M} + \mathcal{M}\mathcal{W}^\top\right]_{ij} = \sum_{k=1}^{2^{n-l}} \mathcal{W}_{ik} + \sum_{k=1}^{2^{n-l}} \mathcal{W}_{jk} \approx \frac{1}{2^l}\, 2 \sum_{q=1}^{2^l}\sum_{k=1}^{2^{n-l}} \mathcal{W}_{qk} \approx \frac{1}{2^l}\, 2Z\;, \tag{C.2}$$

where we used that the average over the lines i and j of the matrix $\mathcal{W}$ should become independent of the line index for large enough matrices. We thus have that

$$\overline{\mathcal{W}}\,\overline{\mathcal{W}}^\top \approx (2^n - 2Z)\,\frac{1}{2^l}\mathcal{H} + \mathcal{W}\mathcal{W}^\top\;. \tag{C.3}$$

The matrix $\mathcal{H}$ has one eigenvalue equal to 2^l, and $2^l - 1$ zero eigenvalues. Therefore, $(2^n - 2Z)\,\frac{1}{2^l}\mathcal{H}$ alone has one eigenvalue $\lambda_0^2 = 2^n - 2Z$, and $2^l - 1$ zero eigenvalues.

Let us discuss the situation when $Z < 2^n/2$, in which case $\lambda_0^2 = 2^n - 2Z > 0$. One can then add $\mathcal{W}\mathcal{W}^\top$ as a perturbation, which lifts up the massive degeneracy without affecting much the non-zero eigenvalue because of the large splitting. Therefore, the non-zero eigenvalues of $\overline{\mathcal{W}}\,\overline{\mathcal{W}}^\top$ should be all the non-zero eigenvalues λ_k^2 of $\mathcal{W}\mathcal{W}^\top$, plus the extra large λ_0^2 eigenvalue. This is the heuristic argument for ($8a$,$8b$), which we support numerically as explained in the text.

We remark that we arrived at the result that the negated problem with weights $\overline{W}(x) = 1 - W(x)$ has one more singular value than the problem with weights $W(x)$ using that the number of solutions of $W(x) = 1$ is less than the number of non-solutions $W(x) = 0$, i.e. $Z < 2^n/2$. This ensured that the $\lambda_0^2 = 2^n - 2Z$ singular value was positive; if it were negative, the perturbative argument should break down, for one would have to restore the positiveness of all the singular values in the end. Therefore what determines which of the two problems, with weights $W(x)$ or $\overline{W}(x)$, has more singular values is which one has more solutions. That we used $Z < 2^n/2$ breaks the symmetry between the two problems, and explains why we cannot use the argument twice doing a double negation $\overline{\overline{W}}(x)$. The argument only goes in one direction, and applies only to the case when $Z < \bar{Z}$.

References

[1] Gavey M R and Johnson D S 1979 *Computers and Intractability: A Guide to the Theory of NP-Completeness* (New York: Freeman)

[2] Arora S and Barak B 2009 *Computational Complexity: A Modern Approach* (New York: Cambridge University).

[3] Valiant L G, *The complexity of enumeration and reliability problems*, 1979 *SIAM J. Comput.* **8** 410-21

[4] Welsh D and Gale A, *The complexity of counting problems*, 2001, in *Aspects of Complexity*, eds Downey R and Hirschfeldt D (Berlin: Walter de Gruyter) p 115-54

[5] Jerrum M R, Valiant L G and Vazirani V V, *Random generation of combinatorial structures from a uniform distribution*, 1986 *Theor. Comput. Sci.* **43** 169-88

[6] Dahlföf V, Jonsson P and Wahlström M, *Counting models for 2SAT and 3SAT formulae*, 2005 *Theor. Comput. Sci.* **332** 265-91.

[7] Fürer M and Kasiviswanathan SP, *Algorithms for counting 2-SAT solutions and colorings with applications*, 2007 *Lect. Notes Comp. Sci.* **4508** pp 47-57

[8] Rényi A, *On measures of entropy and information*, 1961 in *Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability* vol 1 (Berkeley: University of California) pp 547-61

[9] There are in total Z non-vanishing matrix elements in $\mathcal{W}$. There are also at most Z lines in $\mathcal{W}$ that do not contain only 0s (the upper bound is reached if there are Z linearly independent lines, each line with a single entry that is equal to 1). Hence, $r \leq \min(Z, 2^l)$. An analogous argument for the columns yields $r \leq \min(Z, 2^{n-l})$. Therefore, $r \leq \min(2^l, 2^{n-l}, Z)$.

[10] Chamon C and Mucciolo ER, *Virtual parallel computing and a search algorithm using matrix product states*, 2012 *Phys. Rev. Lett.* **109** 030503

[11] Goerdt A, *A threshold for unsatisfiability*, 1996 J. Comput. Syst. Sci. **53** pp 469-86

[12] Chvàtal V and Reed B, *Mick gets some (the odds are on his side)*, 1992 *Proc. 33rd IEEE Symp. on Foundations of Computer Science* (Los Alamitos: IEEE Society) pp 620-7

[13] Monasson R and Zecchina R, *Entropy of the K-satisfiability problem*, 1996 *Phys. Rev. Lett.* **76** 3881-4

[14] Vazirani V V, *Counting DNF solutions*, 2001 *Approximate Algorithms* (Berlin: Springer-Verlag) p 295